Chapter 4 roadmap

4.1 Introduction and Network Service Models
4.2 VC and Datagram Networks
4.3 What’s Inside a Router
4.4 The Internet (IP) Protocol - IPv4, IPv6
4.5 Routing Algorithms
   └ Link state routing
   └ Distance vector routing
   └ Hierarchical Routing
4.6 Routing in the Internet
4.7 Broadcast and Multicast Routing

Hierarchical Routing

Our routing study thus far - idealization

/> all routers identical
/> network "flat"
... not true in practice

scale: with 200 million destinations:
/> can’t store all dest’s in routing tables!
/> routing table exchange would swamp links!

administrative autonomy
/> internet = network of networks
/> each network admin may want to control routing in its own network

aggregate routers into regions, “autonomous systems” (AS)
routers in same AS run same routing protocol
   └ “intra-AS” routing protocol
routers in different AS can run different “intra-AS” routing protocol
gateway routers
/> special routers in AS
/> run intra-AS routing protocol with all other routers in AS
/> also responsible for routing to destinations outside AS
/> run “inter-AS” routing protocol with other gateway routers
Intra-AS and Inter-AS routing

Gateways:
- perform inter-AS routing amongst themselves
- perform intra-AS routing with other routers in their AS

- inter-AS, intra-AS routing in gateway A.c

We'll examine specific inter-AS and intra-AS Internet routing protocols shortly

Chapter 4 roadmap
4.1 Introduction and Network Service Models
4.2 VC and Datagram Networks
4.3 What's Inside a Router
4.4 The Internet (IP) Protocol
  4.4.1 IPv4 addressing
  4.4.2 Moving a datagram from source to destination
  4.4.3 Datagram format
  4.4.4 IP Fragmentation
  4.4.5 ICMP: Internet Control Message Protocol
  4.4.6 DHCP: Dynamic Host Configuration Protocol
  4.4.7 NAT: Network Address Translation
4.5 Routing Algorithms
4.6 Routing in the Internet
4.7 Broadcast and Multicast Routing
The Internet Network layer

Host, router network layer functions:

- **Routing protocols**
  - path selection
  - RIP, OSPF, BGP

- **IP protocol**
  - addressing conventions
  - datagram format
  - packet handling conventions

- **ICMP protocol**
  - error reporting
  - router "signaling"

Transport layer: TCP, UDP

Link layer

Physical layer

---

**IP Addressing: introduction**

- IP address: 32-bit identifier for host, router interface
- Interface: connection between host/router and physical link
  - router typically has multiple interfaces
  - host may have multiple interfaces
  - IP addresses associated with each interface

**IP Addressing**

- IP address:
  - network part (high order bits)
  - host part (low order bits)
- What's a network?
  - from IP address perspective
    - device interfaces with same network part of IP address
    - can physically reach each other without intervening router

Network consisting of 3 IP networks
(for IP addresses starting with 223, first 24 bits are network addresses)
**IP Addressing**

How to find the networks?
- Detach each interface from router, host
- Create "islands of isolated networks"

**Interconnected system consisting of six networks**

223.1.1.1
223.1.1.2
223.1.1.3
223.1.1.4
223.1.1.5
223.1.1.6

223.1.2.1
223.1.2.2
223.1.2.3
223.1.2.4
223.1.2.5
223.1.2.6

223.1.3.1
223.1.3.2
223.1.3.3
223.1.3.4
223.1.3.5
223.1.3.6

223.1.4.1
223.1.4.2
223.1.4.3
223.1.4.4
223.1.4.5
223.1.4.6

223.1.5.1
223.1.5.2
223.1.5.3
223.1.5.4
223.1.5.5
223.1.5.6

223.1.6.1
223.1.6.2
223.1.6.3
223.1.6.4
223.1.6.5
223.1.6.6

**IP Addresses**

given notion of "network", let's re-examine IP addresses:
"class-full" addressing:

<table>
<thead>
<tr>
<th>class</th>
<th>network</th>
<th>host</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.0.0.0 to 127.255.255.255</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>128.0.0.0 to 191.255.255.255</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>192.0.0.0 to 223.255.255.255</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>224.0.0.0 to 239.255.255.255</td>
<td></td>
</tr>
</tbody>
</table>

32 bits

**IP addressing: CIDR**

- **Classful addressing:**
  - inefficient use of address space, address space exhaustion
  - e.g., class B net allocates enough addresses for 65K hosts, even if only 2K hosts in that network; whereas class C allows 254 hosts

- **CIDR: Classless InterDomain Routing**
  - network portion of address of arbitrary length
  - address format: a.b.c.d/x, where x is # bits in network portion of address

<table>
<thead>
<tr>
<th>network part</th>
<th>host part</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000</td>
<td>00010111</td>
</tr>
<tr>
<td>00010000</td>
<td>00000000</td>
</tr>
<tr>
<td>200.23.16.0/23</td>
<td></td>
</tr>
</tbody>
</table>
IP addresses: how to get one?

**Q:** How does host get IP address?

- hard-coded by SysAdmin in a file
  - Wintel: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from AS server
  - "plug-and-play"
  (more shortly…)

IP addresses: how to get one?

**Q:** How does network get network part of IP addr?

**A:** gets allocated portion of its provider ISP’s address space

<table>
<thead>
<tr>
<th>ISP’s block</th>
<th>11001000 00010111 00010000 00000000 200.23.16.0/20</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISP decides…</td>
<td>11001000 00010111 00010000 00000000 200.23.16.0/23</td>
</tr>
<tr>
<td>Organization 0</td>
<td>11001000 00010111 00010000 00000000 200.23.16.0/23</td>
</tr>
<tr>
<td>Organization 1</td>
<td>11001000 00010111 00010100 00000000 200.23.18.0/23</td>
</tr>
<tr>
<td>Organization 2</td>
<td>11001000 00010111 00010100 00000000 200.23.18.0/23</td>
</tr>
<tr>
<td>Organization 7</td>
<td>11001000 00010111 00011110 00000000 200.23.30.0/23</td>
</tr>
</tbody>
</table>

Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information:
- Organization 1 keeps the IP addresses 200.23.18.0/23 and ISPs-R-Us advertises the 200.23.18.0/23 block.
- Fly-by-night-ISP continues to advertise 200.23.16.0/20 – longest prefix matching is used by routers to route to organization 1 via ISPs-R-Us.
- "Send me anything with addresses beginning 200.23.18.0/23"
- "Send me anything with addresses beginning 199.31.0.0/16"
Hierarchical addressing: more specific routes

ISPs-R-Us has a more specific route to Organization 1

```
Organization 0
  200.23.16.0/23

Organization 2
  200.23.20.0/23

Organization 3
  200.23.30.0/23

ISPs-R-Us
```

“Send me anything with addresses beginning 200.23.16.0/20”

```
200.23.18.0/23
200.23.20.0/23
```

Fly-By-Night-ISP

Organization 0

Organization 7

Internet

“Send me anything with addresses beginning 199.31.0/16 or 200.23.16.0/23”

```
200.23.20.0/23
```

```
Organization 1
  199.31.0.0/16

Organization 2

... ...
```

IP addressing: the last word...

Q: How does an ISP get block of addresses?
A: ICANN: Internet Corporation for Assigned Names and Numbers
- allocates addresses
- manages DNS
- assigns domain names, resolves disputes

Moving a datagram from source to dest.

<table>
<thead>
<tr>
<th>IP datagram:</th>
<th>forwarding table in A</th>
<th>Dest. Net.</th>
<th>next router</th>
<th>Nhops</th>
</tr>
</thead>
<tbody>
<tr>
<td>fields</td>
<td>223.1.1</td>
<td>223.1.2</td>
<td>223.1.3</td>
<td></td>
</tr>
<tr>
<td>misc fields</td>
<td></td>
<td>223.1.1.4</td>
<td>223.1.1.4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>223.1.1.4</td>
<td>223.1.1.4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>223.1.1.4</td>
<td>223.1.1.4</td>
<td>2</td>
</tr>
</tbody>
</table>

misc fields:
- datagram remains unchanged, as it travels source to destination
- addr fields of interest here
Getting a datagram from source to dest.

Starting at A, send IP datagram addressed to B:
- look up net. address of B in forwarding table
- find B is on same net. as A
- link layer will send datagram directly to B inside link-layer frame
  - B and A are directly connected

Dest. Net.  next router  Nhops
223.1.1          223.1.1.4  1
223.1.2          223.1.1.4  2
223.1.3          223.1.1.4  2

Getting a datagram from source to dest.

Starting at A, dest. E:
- look up network address of E in forwarding table
- E on different network
  - A, E not directly attached
  - routing table: next hop router to E is 223.1.1.4
- link layer sends datagram to router 223.1.1.4 inside link-layer frame
- datagram arrives at 223.1.1.4
- continued...

Getting a datagram from source to dest.

Arriving at 223.1.4, destined for 223.1.2.2:
- look up network address of E in router's forwarding table
- E on same network as router's interface 223.1.2.9
  - router, E directly attached
  - link layer sends datagram to 223.1.2.2 inside link-layer frame via interface 223.1.2.9
- datagram arrives at 223.1.2.2 (hooray!)
### IP datagram format

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version/Length</td>
<td>The version number and the length of the IP header.</td>
</tr>
<tr>
<td>Type of Service</td>
<td>The type of service the datagram is providing.</td>
</tr>
<tr>
<td>Source/Destination</td>
<td>The source and destination IP addresses.</td>
</tr>
<tr>
<td>Identifier/Hop Limit</td>
<td>The fragment identifier and the maximum hop limit.</td>
</tr>
<tr>
<td>Time to Live</td>
<td>The time to live, which is used to detect invalid routes.</td>
</tr>
<tr>
<td>Protocol</td>
<td>The protocol number of the upper layer protocol.</td>
</tr>
<tr>
<td>Source Port/ Destination Port</td>
<td>The source and destination port numbers for TCP/UDP.</td>
</tr>
<tr>
<td>Options</td>
<td>Additional options that can be added to the datagram.</td>
</tr>
</tbody>
</table>

### IP Fragmentation & Reassembly

- Network links have MTU (max transfer size), which is the largest possible link-level frame.
- Different link types, different MTUs.
- Large IP datagram divided ("fragmented") within the network, resulting in several fragments.
- "Reassembled" only at the final destination.
- IP header bits used to identify, order, and relate fragments.

#### Example

- **4000 byte datagram**
- **MTU = 1500 bytes**

  - Fragmentation: one large datagram into 3 smaller datagrams
  - Reassembly:
    - ID = x, offset = 0, fragflag = 0, length = 4000
    - ID = x, offset = 0, fragflag = 1, length = 1500
    - ID = x, offset = 1480, fragflag = 1, length = 1500
    - ID = x, offset = 2960, fragflag = 0, length = 1040

**Comments:**

- All data link protocols supported by IP must have MTU = min 576 bytes.
- If TCP segment size MSS = 536 bytes, then fragmentation can be completely avoided.